Gianluca Di Tuccio gianluca.dituccio@studio.unibo.it
I choose for the Model 1 --> Decision Tree, while Model 2 --> Gaussian Naive Bayes
The url for uploading the dataset (assuming in the same folder) is 'exam2022_01_13.csv'

1. Load the data and explore them, showing size, structure and histograms of numeric data; show the histogram of the frequencies of the class labels, contained in the "language" column

In [1]:
```
import pandas as pd
url = 'exam2022_01_13.csv'
df = pd.read_csv(url)
df # this is for having an overview of the csv document
```

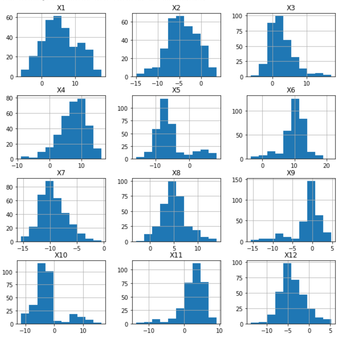
Out[1]:

	language	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12
0	ES	7.071476	-6.512900	7.650800	11.150783	-7.657312	12.484021	-11.709772	3.426596	1.462715	-2.812753	0.866538	-5.244274
1	ES	10.982967	-5.157445	3.952060	11.529381	-7.638047	12.136098	-12.036247	3.491943	0.595441	-4.508811	2.332147	-6.221857
2	ES	7.827108	-5.477472	7.816257	9.187592	-7.172511	11.715299	-13.847214	4.574075	-1.687559	-7.204041	-0.011847	-6.463144
3	ES	6.744083	-5.688920	6.546789	9.000183	-6.924963	11.710766	-12.374388	6.169879	-0.544747	-6.019237	1.358559	-6.356441
4	ES	5.836843	-5.326557	7.472265	8.847440	-6.773244	12.677218	-12.315061	4.416344	0.193500	-3.644812	2.151239	-6.816310
...	...	...	...	...	...	...	...	...	...	...	...	...	...
324	US	-0.525273	-3.868338	3.548304	1.496249	3.490753	5.849887	-7.747027	9.738836	-11.754543	7.129909	0.209947	-1.946914
325	US	-2.094001	-1.073113	1.217397	-0.550790	2.666547	7.449942	-6.418064	10.907098	-11.134323	6.728373	2.461446	-0.026113
326	US	2.116909	-4.441482	5.350392	3.675396	2.715876	3.682670	-4.500850	11.798565	-12.031005	7.566142	-0.606010	-2.245129
327	US	0.299616	0.324844	3.299919	2.044040	3.634828	6.693840	-5.676224	12.000518	-11.912901	4.664406	1.197789	-2.230275
328	US	3.214254	-3.135152	1.122691	4.712444	5.926518	6.915566	-5.799727	10.858532	-11.659845	NaN	0.349482	-5.983281

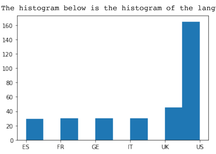
329 rows × 13 columns

In [2]:
```
print(df.shape) # it shows the number of rows and columns
print()
import matplotlib.pyplot as plt
print("The histograms below are refered to the datas:")
pd.DataFrame.hist(df, figsize=[10,10]);
plt.show()
print()
print("The histogram below is the histogram of the language column:")
plt.hist(df['language']);
plt.show()
```

```
(329, 13)

The histograms below are refered to the datas:
```




```
The histogram below is the histogram of the language column:
```



2. Drop the rows with NaN values, if any, show the shape of the dataset after this cleaning

In [3]:
```
df1 = df.dropna()
df1.shape # previously, there are 329 rows
```

Out[3]:
```
(321, 13)
```

3. Tune the hyper–parameters of Model1 (Decision Tree) with Cross Validation on the training set, optimize for recall_macro

Model 1: Decision Tree, optimized for recall_macro

In [4]:
```
from sklearn.model_selection import train_test_split
X = df1.drop(['language'], axis = 1)
y = df1['language']
rnd_state = 10  # it uses for the seed of this program (for having the same simulation results)
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state = rnd_state)
```

In [5]:
```
from sklearn.model_selection import GridSearchCV
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, plot_confusion_matrix

# Decision Tree
param_dt = [{'max_depth':list(range(1,20))}]
# for the cross validation I use a GridSearchCV instead a loop, with cv = 5
clf_dt = GridSearchCV(DecisionTreeClassifier(), param_grid = param_dt,
                      scoring = 'recall_macro', return_train_score = False, cv = 5, n_jobs = -1)
clf_dt.fit(X_train, y_train)
```

Out[5]:
```
GridSearchCV(cv=5, estimator=DecisionTreeClassifier(), n_jobs=-1,
             param_grid=[{'max_depth': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,
                                        13, 14, 15, 16, 17, 18, 19]}],
             scoring='recall_macro')
```

4. Produce a classification report for Model1 (Decision Tree) on the test set

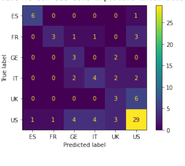
In [6]:
```
from sklearn.metrics import classification_report
y_true, y_pred = y_test, clf_dt.predict(X_test)
print(classification_report(y_true, y_pred))
```

```
              precision    recall  f1-score   support

          ES       0.86      0.86      0.86         7
          FR       0.75      0.38      0.50         8
          GE       0.30      0.60      0.40         5
          IT       0.44      0.40      0.42        10
          UK       0.30      0.33      0.32         9
          US       0.71      0.69      0.70        42

    accuracy                           0.59        81
   macro avg       0.56      0.54      0.53        81
weighted avg       0.62      0.59      0.60        81
```

5. produce the confusion matrix for Model1 (Decision Tree) on the test set

In [7]:
```
from sklearn.metrics import plot_confusion_matrix
plot_confusion_matrix(clf_dt, X_test, y_true);
print('Here we can see some important information, combined with the previous f1 score, recall and precision.')
```

```
Here we can see some important information, combined with the previous f1 score, recall and precision.
```



6. Tune the hyper–parameters of Model2 (Gaussian Naive Bayes) with Cross Validation on the training set, optimize for recall_macro

In [8]:
```
from sklearn.model_selection import GridSearchCV
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, plot_confusion_matrix

param_nb = [{'var_smoothing':[10, 1, 1e-1, 1e-2, 1e-3, 1e-4, 1e-5, 1e-6, 1e-7, 1e-8, 1e-9, 1e-10]}]
clf_nb = GridSearchCV(GaussianNB(), param_grid = param_nb,
                      scoring = 'recall_macro', return_train_score = False, cv = 5, n_jobs = -1)
clf_nb.fit(X_train, y_train)
```

Out[8]:
```
GridSearchCV(cv=5, estimator=GaussianNB(), n_jobs=-1,
             param_grid=[{'var_smoothing': [10, 1, 0.1, 0.01, 0.001, 0.0001,
                                            1e-05, 1e-06, 1e-07, 1e-08, 1e-09,
                                            1e-10]}],
             scoring='recall_macro')
```

7. Produce a classification report for Model2 (Gaussian Naive Bayes) on the test set

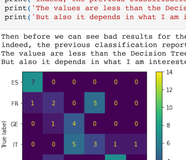
In [9]:
```
y_true, y_pred = y_test, clf_nb.predict(X_test)
print(classification_report(y_true, y_pred))
```

```
              precision    recall  f1-score   support

          ES       0.54      1.00      0.70         7
          FR       0.25      0.25      0.25         8
          GE       0.21      0.80      0.33         5
          IT       0.38      0.30      0.33        10
          UK       0.39      0.78      0.52         9
          US       0.93      0.33      0.49        42

    accuracy                           0.46        81
   macro avg       0.45      0.58      0.44        81
weighted avg       0.66      0.46      0.46        81
```

8. Produce the confusion matrix for Model2 (Gaussian Naive Bayes) on the test set

In [10]:
```
plot_confusion_matrix(clf_nb, X_test, y_true);
print('Then before we can see bad results for the majority of the languages;')
print('indeed, the previous classification report tell us some information about f1 score.')
print('The values are less than the Decision Tree Model.')
print('But also it depends in what I am interested, if f1 or only precision or recall.')
```

```
Then before we can see bad results for the majority of the languages;
indeed, the previous classification report tell us some information about f1 score.
The values are less than the Decision Tree Model.
But also it depends in what I am interested, if f1 or only precision or recall.
```



In [ ]:

In [ ]: